

# Simulación de Proyectos de Software desarrollados con XP: Subsistema de Desarrollo de Tareas

Tamara Kasiak<sup>1,a</sup>, Diego Alberto Godoy<sup>1,b</sup>

<sup>1</sup>Centro de Investigación en Tecnologías de la Información y Comunicaciones  
(C.I.T.I.C.)/Universidad Gastón Dachary  
Salta 1968. Posadas, Misiones, Argentina  
Teléfono: +54-376-4438677

<sup>a</sup>tamarakasiak@ugd.edu.ar; <sup>b</sup>diegodoy@ugd.edu.ar

## Resumen

Administrar Proyectos de Software siguiendo Programación Extrema (XP) implica implementar, de forma conjunta y al extremo, prácticas ya conocidas en el ámbito del Desarrollo de Software, lo que torna a esta actividad aún más compleja. Como una manera de tratar esta complejidad, es que se ha construido un Modelo Dinámico de Simulación (siguiendo los lineamientos de la Dinámica de Sistemas), que agrupa las variables involucradas en un proyecto llevado a cabo con XP, que permite analizar el efecto de la implementación conjunta de las prácticas de dicha metodología y ayuda en la gestión de este tipo de proyectos. El modelo presentado en este trabajo corresponde al Subsistema de Desarrollo de Tareas, uno de los más importantes. Este modelo se ha validado con datos de dos proyectos reales. Además, se han diseñado y ejecutado una serie de experimentos sobre el mismo y se ha realizado el Análisis de Sensibilidad de sus variables más importantes. El modelo construido sirve como ayuda a administradores de proyectos novatos, permitiéndoles conocer de antemano las consecuencias de sus decisiones.

**Palabras claves:** Administración de Proyectos de Software, Programación Extrema, Dinámica de Sistemas.

## Contexto

El trabajo presentado aquí se realizó en el marco de la línea de investigación denominada “Simulación como herramienta para la Mejora de los Procesos de Software” iniciada hace dos años en el C.I.T.I.C. Como uno de los resultados de esta línea se presenta parte del

Trabajo de Grado titulado “Modelo de Simulación Dinámico de Gestión de Proyectos de Desarrollo de Software que utilizan Programación Extrema”, de la carrera Licenciatura en Sistemas de Información de la Universidad Gastón Dachary.

## Introducción

En el ámbito de la Ingeniería del Software la evolución de las Metodologías de Desarrollo de Software ha llevado a la aparición de las denominadas Metodologías Ágiles, las cuales están destinadas a romper con la rigidez de las Tradicionales, caracterizadas por la extensa documentación del proceso de desarrollo y por la inflexibilidad ante los cambios.

Una de las Metodologías Ágiles más importante y reconocida es la Programación Extrema (XP) [1], para aquellos proyectos de software donde el cambio en los requerimientos es la norma. Por tal motivo reúne un conjunto de prácticas sencillas ya conocidas, pero que en este caso son llevadas a cabo conjuntamente y en forma extrema. Debido a esto, la gestión de proyectos de desarrollo de software se torna algo impredecible y compleja y es difícil anticipar por sus administradores, los efectos que tiene sobre la marcha del mismo, la aplicación de las diferentes prácticas de XP.

No obstante, si bien existen actualmente diversas herramientas destinadas a ayudar a los administradores de proyectos en la estimación y toma de decisiones, como ser los Modelos Dinámicos de Simulación, que permiten evaluar diferentes alternativas de decisión en la gestión, sin intervenir en el desarrollo real del proyecto, la mayoría de ellos han

sido desarrollados para proyectos que se llevan a cabo con Metodologías Tradicionales, como ser el modelo de Abdel-Hamid y Madnick [2] y el Modelo Dinámico Reducido [3].

El modelo de Abdel-Hamid y Madnick, que data de los años 90, no tiene en cuenta las fases de Definición de Requisitos, Operación y Mantenimiento, por lo que se centra en el Diseño, Codificación, Revisión, Corrección y Pruebas, considerando que los requisitos del proyecto se mantienen estables a lo largo del ciclo de vida. Por otro lado, si bien este modelo está dividido en cuatro subsistemas complejos (Gestión de Recursos Humanos, Producción de Software, Control y Planificación), este está destinado a simular proyectos que se llevan a cabo bajo Metodologías Tradicionales, ya que en la época de su surgimiento aún no se hablaba de Metodologías Ágiles ni de cambios de requerimientos durante el desarrollo.

Lo mismo ocurre para el caso del Modelo Dinámico Reducido, el cual constituye una reducción del Modelo de Abdel-Hamid y Madnick. Los dos criterios de reducción más importantes, en los que se basó este modelo son por un lado, agregar las variables de nivel más relevantes del Modelo de Abdel-Hamid y Madnick y, por otro, elegir los bucles de realimentación adecuados a las variables de nivel elegidas.

## Objetivos del Modelo Propuesto

De acuerdo con lo anterior, se planteó en este trabajo la construcción de un “Modelo de Simulación Dinámico de Gestión de Proyectos de Desarrollo de Software que utilizan XP”, con el objetivo de analizar el efecto que tiene el uso de las prácticas de XP en la gestión de proyectos de desarrollo de software.

El modelo permite, a los administradores de proyectos, evaluar el impacto de sus decisiones de gestión sobre el mismo, a lo largo del tiempo, como así también posibilita comparar los resultados de sus decisiones, ofreciendo datos

suficientes para elegir la mejor opción a ser aplicada en el sistema real.

Esto es posible ya que el modelo habilita a los administradores a realizar cambios en variables críticas del proyecto, como ser recursos, tiempo, tareas a desarrollar, etc., y observar las repercusiones de dichos cambios en el resto del proyecto, y lo más importante, sin comprometer la ejecución real del mismo.

Como se mencionó, el modelo refleja el efecto del uso de prácticas XP en proyectos de desarrollo de software y para ello, su estructura permite simular el desarrollo de una versión o entrega de un proyecto XP a la vez. Es decir, como XP presenta un estilo de desarrollo iterativo e incremental, dentro de un proyecto se negocian con el cliente varias entregas o versiones, por lo que el modelo, está destinado a simular las versiones de a una por vez.

## Construcción del Modelo

Para la construcción del modelo se utilizó el software VenSim PLE 5.4c (Versión Académica) [4] y se han seguido, las etapas de la Metodología de Dinámica de Sistemas [5], de esta forma, en la Fase de Conceptualización se ha construido el Diagrama Causal, incluyendo las variables que representan un proyecto de software XP y sus interrelaciones. Luego, en la Fase de Formulación se ha traducido el Diagrama anteriormente mencionado al Diagrama de Forrester, el cual ha sido dividido en Subsistemas Conservativos de acuerdo con [6]. Finalmente, en la Fase de Evaluación se han realizado las corridas de validación y experimentales del modelo.

En la Figura 1, se presenta uno de los subsistemas más importantes del modelo: El Subsistema de Desarrollo de Tareas.

Este Subsistema representa todas las etapas por las que pasan cada una de las tareas, en que se dividen las historias de usuario (HU), comenzando con la planificación de dichas tareas para cada iteración, la cual se basa en la cantidad de “HU planificadas por iteración” y en la “Cantidad promedio de tareas por HU”.

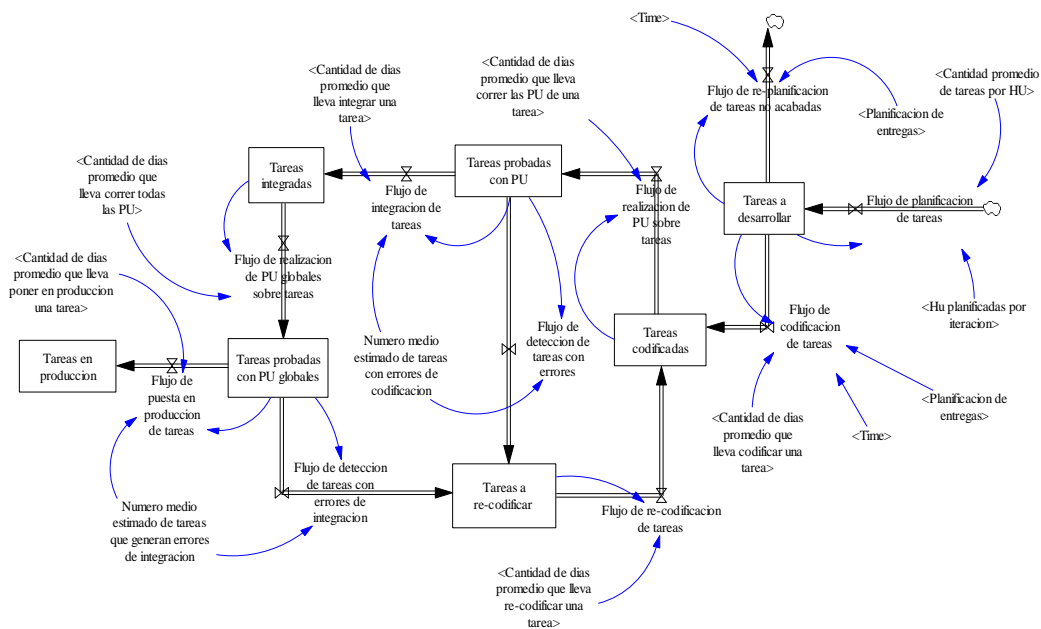


Figura 1 – Subsistema de Desarrollo de Tareas

De esta forma, una vez planificadas las tareas, se encuentran inicialmente en el nivel “Tareas a desarrollar”, y luego van pasando por todos los niveles hasta llegar al nivel “Tareas en Producción”, el cual representa la cantidad de tareas desarrolladas en cada iteración.

Además del subsistema presentado, el modelo se compone de los siguientes subsistemas: Recursos Humanos, Factor de Carga, Presión en el Plazo, Desarrollo de Pruebas de Unidad, Desarrollo de Pruebas de Aceptación, Planificación, Desarrollo de Tareas Extras, Desarrollo de Historias de Usuario, Horas de Trabajo por día y Actualización de los Días Ideales de Ingeniería por Historia de Usuario y por Tarea.

## Validación

En la validación se utilizaron datos reales de dos Proyectos realizados con XP.

El primero es un proyecto de Desarrollo de un Sistema de Gestión de una Empresa de Confecciones, cuyos detalles se pueden ver en [7].

El segundo fue el Sistema “ONess”, un Proyecto Open Source para el “Negocio Textil Mayorista” [8]. Este, consistió en una versión (dividida en 4 iteraciones), que fue

desarrollada en 85 días por 1 sólo programador. Los datos de las iteraciones, de este caso se pueden ver en el Cuadro 1.

Iteración	Historias de Usuario (HU)	Tareas en Promedio por HU	Duración Real en días
1	4	12	45
2	3	9	10
3	4	12	20
4	3	9	10

Cuadro 1 – Iteraciones del proyecto presentado en [8]

De acuerdo con los datos del Cuadro 1, el Gráfico 1 refleja el Flujo de Planificación de Tareas, que también pertenece al subsistema de la Figura 1.

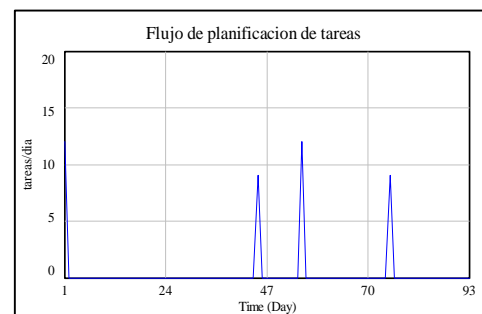


Gráfico 1 - Flujo de Planificación de Tareas

Por otro lado, en el Gráfico 2 se puede ver que el modelo se comportó de acuerdo con los datos del caso de validación, alcanzándose el nivel de Tareas

en Producción planificado para cada iteración (Gráfico 1).

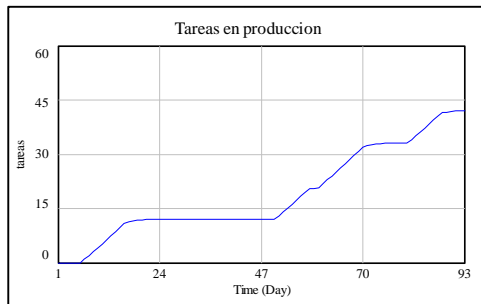


Gráfico 2 – Tareas en Producción

## Experimentos realizados

Al igual que en la validación, el modelo ha sido simulado bajo diversas condiciones que representan situaciones típicas en proyectos XP. En este caso se presenta uno de los experimentos realizados, el cual se refiere al desarrollo de una versión de un proyecto XP, la cual engloba 15 HU a ser completadas en 45 días. Esta versión fue dividida en 3 iteraciones de 5 HU cada una. Así también, del desarrollo de la misma participaron 6 pares de programadores (4 novatos y 8 expertos).

En este escenario se analizó la incorporación de 3 HU más al final de la primera iteración, el abandono del proyecto de 2 programadores al comienzo de la segunda iteración y la re-planificación de HU no acabadas al final de primera y segunda iteración.

De esta forma, en el Gráfico 3 se puede observar como aumenta el tamaño de la versión, de 15 para 18 HU, al actualizarse la cantidad de HU al final de la primera iteración (día 14).

Por otro lado, en el Gráfico 4 se puede observar como la cantidad de Pares de Programadores disminuye el día 15 (comienzo de la segunda iteración), cuando 2 programadores abandonan el equipo de desarrollo. Debido a esta situación, el crecimiento en la cantidad de Programadores Expertos (que aumenta a medida que los novatos adquieren experiencia) se ve interrumpido el día 15, como se puede apreciar en el Gráfico 5.

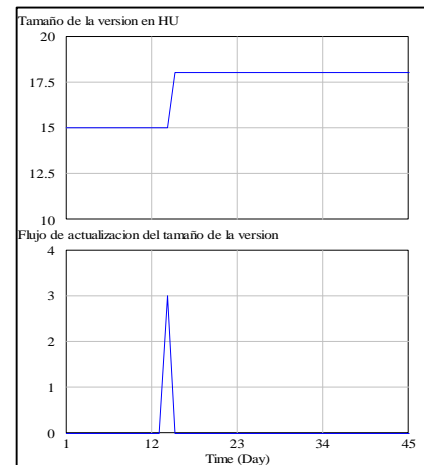


Gráfico 3 – Actualización del Tamaño de la Versión

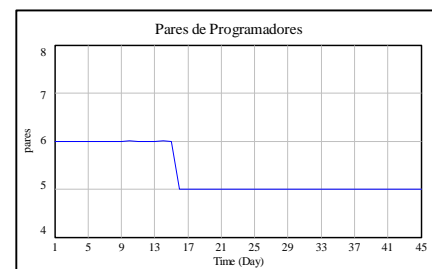


Gráfico 4 – Cantidad de Pares de Programadores

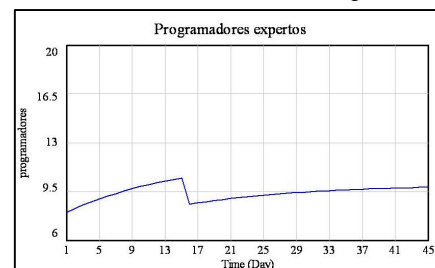


Gráfico 5 – Cantidad de Programadores Expertos

Finalmente, en el Gráfico 6 se refleja la replanificación de HU no acabadas al final de las 2 primeras iteraciones. La replanificación en la iteración 1 corresponde al hecho de que, como se observa en el Gráfico 5, la cantidad de programadores expertos es baja al comienzo de la versión, y por lo tanto, la cantidad de horas de trabajo diarias disponibles para desarrollo es también baja en esta etapa.

La replanificación en la iteración 2 corresponde, en cambio, al incremento en la cantidad de HU planificadas para dicha iteración (Gráfico 3) y al abandono del proyecto por parte de 2 programadores tal como se mostró en el Gráfico 4.

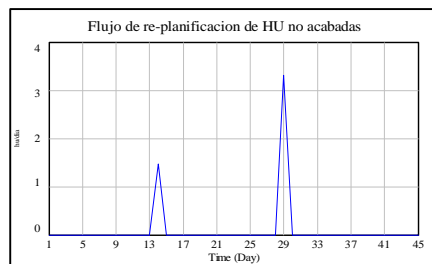


Gráfico 6 – Replanificación de HU no acabadas

## Resultados

En el presente trabajo se ha construido un “Modelo de Simulación Dinámico de Gestión de Proyectos de Desarrollo de Software que utilizan XP”, lo cual hace que se diferencie de otros trabajos en los que se modelan proyectos con Metodologías Tradicionales. De esta forma, el modelo puede ser usado como herramienta para analizar el efecto que tiene el uso de prácticas de XP en proyectos de software. En este sentido, es preciso mencionar que las prácticas que quedaron reflejadas en el modelo, de acuerdo Kent Beck en [9], son las siguientes: “El juego de la planificación”, “Versiones pequeñas”, “Hacer pruebas”, “Recodificación”, “Programación en parejas”, “Integración continua” y “40 horas semanales”, que son las que fueron posibles de representar utilizando Dinámica de Sistemas.

Para ofrecer una buena flexibilidad el modelo permite la alteración de valores durante su ejecución, como ser la cantidad de requerimientos del cliente, las fechas de comienzo y entrega de cada iteración, la cantidad de requerimientos a ser desarrollados en cada iteración, la cantidad de programadores, las horas extras agregadas por día, el porcentaje de tiempo destinado a cada etapa del desarrollo, el factor de carga del equipo de desarrollo, entre otras.

La validación del modelo realizada con los proyectos [7] y [8] como casos de “entrenamiento” fue positiva ya que el mismo se comportó de acuerdo con los datos reales. Finalmente, luego de validar el modelo, ejecutarlo bajo diferentes escenarios y realizar el análisis de sensibilidad, se ha llegado a la conclusión de que el mismo cumple con sus objetivos y puede ser utilizado como herramienta para evaluar diferentes decisiones de gestión

sobre proyectos de software desarrollados con XP.

## Trabajos Futuros

Como trabajos futuros se plantea la posibilidad de lograr desarrollar un modelo íntegro que permita simular el desarrollo completo de un Proyecto, es decir, con todas sus versiones a la vez.

Una variante que podría ser incorporada al modelo es la “Planificación por alcance” dentro de la Metodología XP ya que el mismo fue construido basándose en la “Planificación por tiempo”.

Además sería interesante la construcción de una jerarquía de Modelos Dinámicos para simular Proyectos llevados a cabo con Metodologías Ágiles. Dicha jerarquía de modelos podría estar inspirada en la presentada en [2] para Metodologías Tradicionales, la cual divide a los modelos en Básicos, Intermedios y Avanzados.

## Formación de Recursos Humanos

El equipo de trabajo se encuentra formado por un investigador Junior y dos estudiantes en período de realización de trabajos finales de grado. El número de tesinas de grado aprobadas es uno y una en proceso.

## Referencias

- [1] <http://www.programacionextrema.org>. Última actualización sábado, 15 de enero de 2005.
- [2] Abdel – Hamid T. K., Madnick S. E.; “Software Project Dynamics An Integrate Approach”, Editorial Prentice –New Jersey, 1991. ISBN:0-13-822040-9 .
- [3] Ramos I., Toro M., Ruiz M.; “Modelo Dinámico Reducido”, Informe Técnico: LSI-2001-01, Universidad de Sevilla, 2001.
- [4] <http://www.vensim.com/index.html>. Última actualización 26/01/2012.
- [5] Aracil, J.; “Dinámica de Sistemas”, Editorial Isdefe, España, 1995. ISBN: 8-46-833802-8
- [6] Torrealdea J.; “Dinámica de Sistemas. Elementos y Estructuras de un Modelo”. Ciencias de la Computación e Inteligencia Artificial. Universidad del País Vasco. País Vasco.
- [7] <http://users.dsic.upv.es/asignaturas/facultad/lsi/eje/mploxp/>. Última actualización 11/01/2004.
- [8] <http://oness.sourceforge.net/proyecto/html/index.html>. Última actualización 29/09/2004
- [9] Beck, K.; “Una Explicación de la Programación Extrema. Aceptar el Cambio”, Editorial Addison Wesley, España, 2002. ISBN: 8-47-829055-9